ggplot

Jiacheng

2022-11-17

library(knitr)  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(tsibble)

##   
## Attaching package: 'tsibble'  
##   
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, union

library(lubridate)

##   
## Attaching package: 'lubridate'  
##   
## The following object is masked from 'package:tsibble':  
##   
## interval  
##   
## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

#library(sf)  
library(tmap)  
library(terra)

## terra 1.6.17  
##   
## Attaching package: 'terra'  
##   
## The following object is masked from 'package:tidyr':  
##   
## extract  
##   
## The following object is masked from 'package:knitr':  
##   
## spin

library(ceramic)  
library(tsibble)  
library(tsibbledata)  
library(ggHoriPlot)  
library(ggstream)  
library(colorspace)

##   
## Attaching package: 'colorspace'  
##   
## The following object is masked from 'package:terra':  
##   
## RGB

library(ggalluvial)  
library(scales)

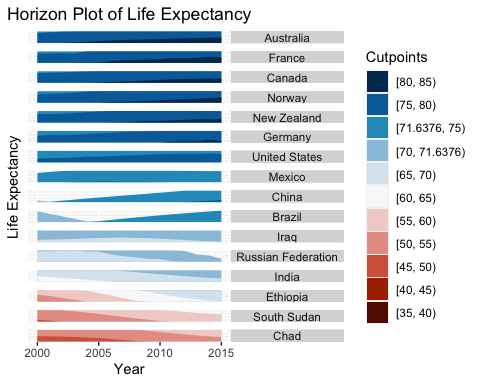
##   
## Attaching package: 'scales'  
##   
## The following object is masked from 'package:terra':  
##   
## rescale  
##   
## The following object is masked from 'package:purrr':  
##   
## discard  
##   
## The following object is masked from 'package:readr':  
##   
## col\_factor

#read data  
life<-read.csv("/Users/yujiacheng/Desktop/Data\_Visualization\_Project/life\_expectancy\_data.csv")  
#rename columns  
colnames(life)[colnames(life)=="Life\_expectancy"] <- "life\_expectancy"  
colnames(life)[colnames(life)=="Adult\_mortality"] <- "adult\_mortality"  
colnames(life)[colnames(life)=="Infant\_death"] <- "infant\_deaths"  
#life

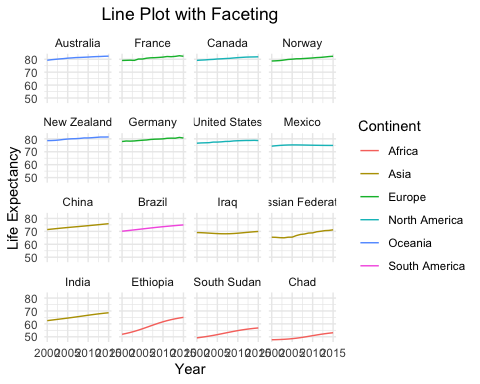
#choose 16 countries from different continents as plot examples  
subset\_country <- c("United States",  
 "China",  
 "Australia",  
 "Brazil",  
 "Canada",  
 "South Sudan",  
 "France",  
 "Germany",  
 "India",  
 "Iraq",  
 "Mexico",  
 "New Zealand",  
 "Norway",  
 "Russian Federation",  
 "Chad",  
 "Ethiopia")

#csv with the countries' continent in order to combine the country with its continent  
continent= read.csv("/Users/yujiacheng/Desktop/Data\_Visualization\_Project/Continents.csv")

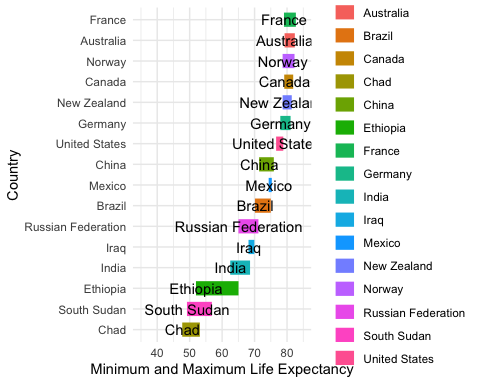
#the horizon plot  
  
#according to the range of life expectancy decide the cutpoints  
cutpoints <- seq(35, 85, by = 5)  
#compute the mean of the life expectancy of these 16 countries  
#life %>%  
 #filter(Country %in% subset\_country) %>%  
 #summarise(mean\_life = mean(life\_expectancy, na.rm = TRUE))  
  
life %>%  
 filter(Country %in% subset\_country) %>%  
 ggplot() +  
 #origin defined as the mean of the life expectancy of these 16 countries  
 geom\_horizon(aes(Year,life\_expectancy, fill = ..Cutpoints..), origin = 71.63764,horizonscale = cutpoints) +  
 scale\_fill\_hcl(palette = 'RdBu') +  
 facet\_grid(reorder(Country, -life\_expectancy) ~ .) +  
 theme(  
 strip.text.y = element\_text(angle = 0),  
 axis.text.y = element\_blank(),  
 axis.ticks.y = element\_blank()  
 )+  
 ggtitle("Horizon Plot of Life Expectancy") +  
 ylab("Life Expectancy")+  
 theme(plot.title = element\_text(hjust = 0.5))



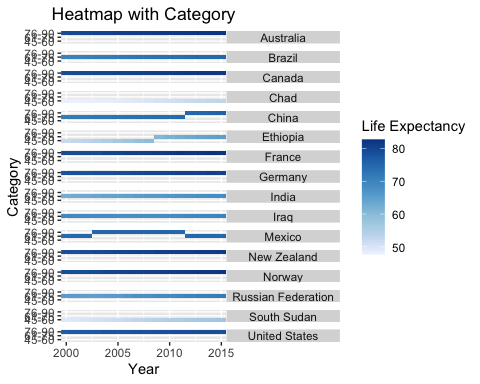
#Line plot is to use faceting with reordering countries by life expectancy  
  
#add a new column to merge continent with each country  
life=merge(continent,life)  
life %>%  
 filter(Country %in% subset\_country) %>%  
 ggplot() +  
 #lines' color by different continent  
 geom\_line(aes(Year,life\_expectancy,color=Continent)) +  
 #facet by different country  
 facet\_wrap(~ reorder(Country, -life\_expectancy))+  
 labs(color="Continent")+  
 theme\_minimal()+  
 ggtitle("Line Plot with Faceting") +  
 ylab("Life Expectancy")+  
 theme(plot.title = element\_text(hjust = 0.5))



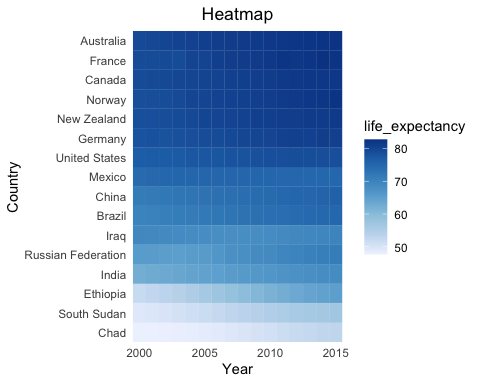
#Segment marks  
  
life %>%  
 filter(Country %in% subset\_country) %>%  
 group\_by(Country) %>%  
 #'Summarise' finds the minimum and maximum life expectancy restricted to each country  
 summarise(  
 min\_life = min(life\_expectancy),  
 max\_life = max(life\_expectancy)  
 )%>%  
 ggplot() +  
 #each bar isn't anchored at 0 due to 'segment' call  
 geom\_segment(  
 #reorder the country(y axis) by maximum life expectancy  
 aes(min\_life, reorder(Country, max\_life), xend = max\_life, yend = Country, col = Country),  
 size = 5,  
 ) +  
 labs(x = "Minimum and Maximum Life Expectancy", col = "Country", y = "Country") +  
 xlim(35, 85)+  
 #add country name on each of the segment  
 geom\_text(  
 aes(x = min\_life, y = reorder(Country, max\_life), label = Country),  
 nudge\_x = 0.08   
 )+  
 theme\_minimal()



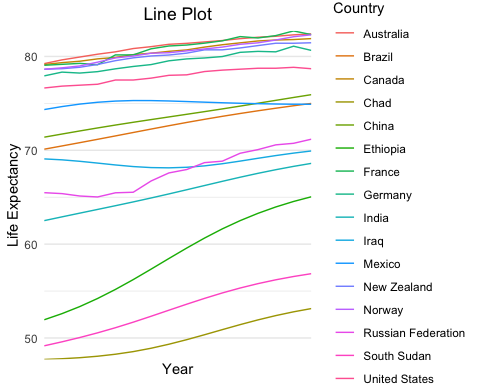
#heatmap with category  
life$category <-cut(life$life\_expectancy,breaks=c(-Inf,60,75,Inf),labels=c("45-60","61-75","76-90"))  
life %>%  
 filter(Country %in% subset\_country) %>%  
 ggplot() +  
 geom\_tile(aes (Year, category, fill = life\_expectancy))+  
 scale\_x\_continuous (expand = c(0, 0))+  
 scale\_fill\_distiller (direction = 1) +  
 facet\_grid(Country ~ .)+  
 theme(strip.text.y = element\_text (angle = 0))+  
 labs(y = "Category",fill='Life Expectancy')+  
 ggtitle("Heatmap with Category") +  
 theme(plot.title = element\_text(hjust = 0.5))



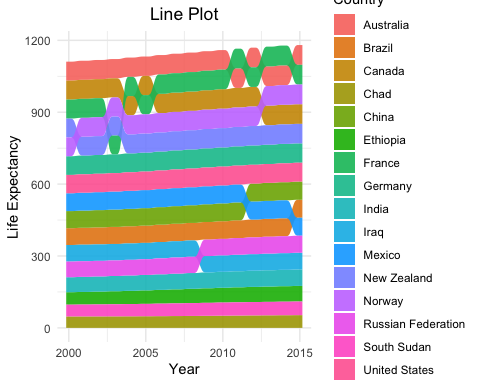
#a heatmap with each country on its own line  
  
life %>%  
 filter(Country %in% subset\_country) %>%  
 ggplot() +  
 geom\_tile(aes(Year, reorder(Country, life\_expectancy), fill = life\_expectancy, col = life\_expectancy)) +  
 scale\_fill\_distiller(direction = 1) +  
 scale\_x\_continuous(expand = c(0, 0))+  
 #scale\_x\_yearquarter(expand = c(0, 0)) +  
 scale\_y\_discrete(expand = c(0, 0)) +  
 scale\_color\_distiller(direction = 1)+  
 labs(y = "Country")+  
 theme\_minimal()+  
 ggtitle("Heatmap") +  
 theme(plot.title = element\_text(hjust = 0.5))



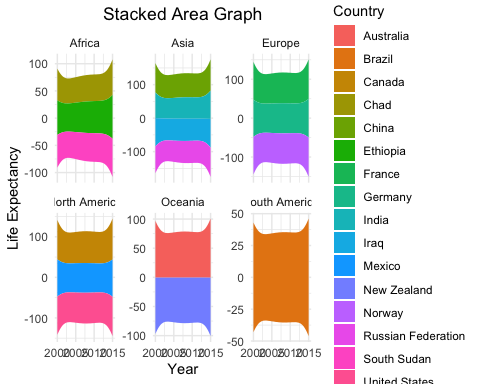
life %>%  
 filter(Country %in% subset\_country)%>%  
 ggplot()+  
 geom\_line(aes(x=Year, y=life\_expectancy, color=Country))+  
 scale\_y\_continuous(expand = c(0, 0))+  
 scale\_x\_discrete(expand = c(0, 0)) +  
 theme(legend.position="bottom")+  
 labs(y = "Life Expectancy")+  
 theme\_minimal()+  
 ggtitle("Line Plot") +  
 theme(plot.title = element\_text(hjust = 0.5))



life %>%  
 filter(Country %in% subset\_country)%>%  
 ggplot()+  
 geom\_alluvium(aes(Year, life\_expectancy, fill = Country, alluvium = Country), decreasing = FALSE, alpha = 0.9) +  
 theme(legend.position = "bottom")+  
 theme\_minimal()+  
 labs(y = "Life Expectancy")+  
 ggtitle("Line Plot") +  
 theme(plot.title = element\_text(hjust = 0.5))



#the stacked area graph  
life=merge(continent,life)  
life %>%  
 filter(Country %in% subset\_country) %>%  
 ggplot() +  
 geom\_stream(aes(Year, life\_expectancy, fill = Country), bw = 1) +  
 facet\_wrap(~ Continent, scale = "free\_y")+  
 labs(y = "Life Expectancy")+  
 theme\_minimal()+  
 ggtitle("Stacked Area Graph") +  
 theme(plot.title = element\_text(hjust = 0.5))



#the alluvial plot to explicitly rank the groups of continents  
life=merge(continent,life)  
life %>%  
 filter(Country %in% subset\_country) %>%  
 ggplot() +  
 geom\_alluvium(aes(Year, life\_expectancy, fill = Country, alluvium = Country), decreasing = FALSE, alpha = 0.9) +  
 facet\_wrap(~ Continent, scale = "free\_y") +  
 scale\_y\_continuous(labels = label\_number(scale\_cut = cut\_long\_scale()))+  
 labs(y = "Life Expectancy")+  
 theme\_minimal()+  
 ggtitle("Alluvial Plot") +  
 theme(plot.title = element\_text(hjust = 0.5))

